Table of Contents

Tóm tắt

Dự án này tập trung vào việc xây dựng và tối ưu hóa hệ thống hỗ trợ chẩn đoán ung thư vú dựa trên bộ dữ liệu Breast Cancer Wisconsin (Diagnostic), được thu thập từ các phép đo kỹ thuật chi tiết qua phương pháp chọc hút kim nhỏ (FNA). Thay vì chỉ dựa vào trực giác lâm sàng, hệ thống đã phân tích các đặc trưng hình thái tế bào như kích thước, độ nén và tính đối xứng để tự động phân loại khối u là lành tính (Benign) hay ác tính (Malignant). Quy trình thực hiện được thiết kế nghiêm ngặt từ khâu xử lý dữ liệu, kiểm soát giá trị ngoại lai đến việc sử dụng RandomizedSearchCV kết hợp StratifiedKFold để tìm ra bộ siêu tham số tối ưu. Qua quá trình thử nghiệm, mô hình Logistic Regression đã được lựa chọn làm giải pháp cuối cùng nhờ khả năng đạt chỉ số Test F1-score lên đến 99.21%, giúp đảm bảo độ tin cậy cực cao trong môi trường y tế.

Giới thiệu

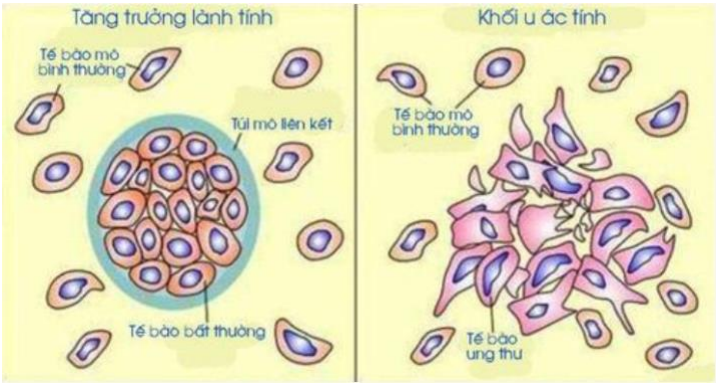
Ung thư vú là một trong những căn bệnh ung thư phổ biến nhất và là nguyên nhân gây tử vong hàng đầu ở phụ nữ trên toàn thế giới. Để đối phó với căn bệnh này, việc chẩn đoán sớm và chính xác đóng vai trò quyết định đến hiệu quả điều trị và khả năng sinh tồn của bệnh nhân. Để có thể góp phần vào việc giúp dự đoán tỷ lệ mắc căn bệnh này, bộ dữ liệu Breast Cancer Wisconsin (Diagnostic), được thu thập bởi các nhà nghiên cứu tại Đại học Wisconsin, Mỹ, sẽ được sử dụng (Wolberg et al. 1995). Dữ liệu này bao gồm các phép đo kỹ thuật chi tiết từ hình ảnh của các tế bào được lấy qua phương pháp chọc hút kim nhỏ (FNA) từ khối u ở vú. Thay vì chỉ dựa vào trực giác lâm sàng, hệ thống sẽ phân tích các đặc trưng hình thái như kích thước, độ nén và tính đối xứng của nhân tế bào để tự động phân loại khối u là Lành tính (Benign) hay Ác tính (Malignant), giúp các bác sĩ có thêm một công cụ hỗ trợ chẩn đoán với độ chính xác và tin cậy cao.

I: Thông tin bộ dữ liệu

1.1: Cách phân biệt khối u lành tính và ác tính

Theo bài báo được đăng trên trang thông tin của bệnh viện Vinmec, việc phân biệt khối u lành tính và ác tính dựa trên hình dạng và đặc tính sinh học là bước then chốt trong chẩn đoán ung thư vú (Vinmec 2024). Các khối u lành tính (Benign) thường có kích thước nhỏ, tốc độ tăng trưởng chậm và sở hữu hình dạng tròn hoặc bầu dục đều đặn. Đặc điểm quan trọng nhất của chúng là ranh giới rõ ràng, bề mặt nhẵn nhụi, ít gồ ghề và không xâm lấn các mô lân cận, giúp việc phẫu thuật loại bỏ trở nên dễ dàng hơn.

Ngược lại, khối u ác tính (Malignant), có thể nghi ngờ mắc bệnh ung thư, thường lớn nhanh và có hình dạng dị dạng, không đối xứng. Bề mặt khối u rất gồ ghề với các đường viền răng cưa hoặc dạng rễ cây cắm sâu vào mô xung quanh, phản ánh tính chất xâm lấn đặc trưng. Dưới kính hiển vi, tế bào ác tính có nhân lớn bất thường, sẫm màu và cấu trúc hỗn loạn. Trong khi khối u lành tính chỉ phát triển tại chỗ, khối u ác tính có khả năng tách rời và di căn đến các cơ quan xa qua hệ máu, gây nguy hiểm trực tiếp đến tính mạng nếu không được điều trị kịp thời.



Hình ảnh 1: Sự khác nhau giữ khối u lành tính và ác tính

1.2: Ý nghĩa của các biến trong bộ dữ liệu

Để hiểu cách mô hình máy học dự đoán ung thư, chúng ta cần nắm rõ ý nghĩa của các chỉ số hình thái tế bào trong bộ dữ liệu Wisconsin. Mỗi biến số đại diện cho một đặc tính vật lý của nhân tế bào được quan sát dưới kính hiển vi:

- Radius (Bán kính): Khoảng cách từ tâm đến các điểm trên đường viền, khối u ác tính thường có tế bào lớn hơn.
- Texture (Kết cấu): Mức độ biến đổi các giá trị thang độ xám trong hình ảnh, phản ánh sự không đồng nhất của bề mặt tế bào.
- Perimeter (Chu vi): Tổng chiều dài đường biên quanh nhân tế bào.
- Area (Diện tích): Không gian mà nhân tế bào chiếm chỗ, diện tích lớn là dấu hiệu nghi vấn ác tính.
- Smoothness (Độ mịn): Sự thay đổi cục bộ của các độ dài bán kính, tế bào lành tính thường có biên độ thay đổi đều đặn hơn.
- Compactness (Độ nén): Chỉ số này cho biết hình dạng của tế báo có "gọn gàng" hay không. Tế bào ung thư có hay có hình dạng kéo dài và phức tạp
- Concavity (Độ lõm): Mức độ nghiêm trọng của các vùng lõm trên đường viền nhân tế bào.
- Concave Points (Số điểm lõm): Đếm số lượng các phần lõm trên đường biên; tế bào ung thư thường có nhiều điểm lồi lõm bất thường.
- Symmetry (Độ đối xứng): Đánh giá mức độ cân đối của nhân tế bào, sự mất đối xứng thường liên quan đến sự phát triển ác tính.
- Fractal Dimension (Kích thước Fractal): Sử dụng lý thuyết "phân mảnh" để đo độ thô ráp của đường biên, giá trị này càng cao thì đường viền tế bào càng phức tạp và gồ ghề.

Mỗi chỉ số trên đều được cung cấp ở ba giá trị: Mean (Trung bình), SE (Sai số chuẩn) và Worst (Giá trị lớn nhất), giúp mô hình có cái nhìn toàn diện từ tổng thể đến những dấu hiệu bất thường nhỏ nhất.

II: Phân tích và khám phá dữ liệu (EDA) và kiểm định dữ liệu (Data Validation)

2.1: Tổng quan thông số thống kê và đặc tính của dữ liệu

2.1.1: Mã hoá biến mục tiêu (Target variable)

Với một số những mô hình máy học (Machine Learning) như Logistic Regressuion dựa trên các phép tính toán học. Nên máy tính không thể xử lý các văn bản như " Benigm" hay "Malignant", do đó ta nên mã hoá chúng thành 1 (Malignant) và 0 (Benigm) để mô hình có thể tính toán xác suất và phân loại chính xác.

2.1.2: Thống kê tổng quát và kiểm định dữ liệu

- Thống kê tổng quát

Bằng cách sử dụng câu lệnh df.describe(), ta có được 1 bảng chứa các thông số thống kê (min,max,mean,std, etc…) của tất cả các biến. Bởi các thống số của các tế bào đều phải lớn hơn 0, nên qua kiểm tra tất cả các giá trị min của các biến, ta biết được các biến đều trong khoảng gía trị hợp lệ. Bên cạnh đó, các thông số về giá trị trung bình (mean) và giá trị lớn nhất (max) cho thấy sự phân hóa rõ rệt giữa các thuộc tính. Cụ thể, các biến như diện tích tế bào có

thể đạt mức tối đa lên tới 2501.0, trong khi các chỉ số về độ mịn của tế bào hay kích thước fractal lại có giá trị rất nhỏ, chỉ dao động quanh ngưỡng 0.1 đến 0.2. Ta cũng thấy được nhưng biến có chữ "mean" – trung bình thường có giá trị nhỏ hơn những biến có chữ "worst" – giá trị tệ nhất, trong tên, là bởi giá trị worst ở đây là giá trị lớn nhất còn mean là giá trị trung bình. Điều này hoàn toàn logic vì giá trị "worst" đại diện cho những đặc điểm tế bào cực đoan nhất được ghi nhận từ mỗi bệnh nhân. Khi so sánh sâu hơn, ta nhận thấy các đặc trưng ác tính thường đẩy các chỉ số "worst" này lên rất cao, tạo ra một khoảng cách lớn so với giá trị trung bình.
Sự chênh lệch cực lớn về thang đo giữa các đơn vị đo lường này là một tín hiệu quan trọng, cho thấy dữ liệu cần được thực hiện chuẩn hóa (Standardization) trước khi đưa vào huấn luyện các mô hình như Logistic Regression. Việc này đảm bảo các biến có giá trị lớn không lấn át các biến có giá trị nhỏ, giúp mô hình nhận diện chính xác các đặc điểm hình thái tinh vi nhất của tế bào ung thư.

- So sáng các giá trị Min, Max với biến mục tiêu (Target Variable)

Trong bộ dữ liệu này, thông thường các giá trị của các biến càng nhỏ thì sẽ có khả năng cao khối u sẽ được dự đoán là lành tính và ngược lại, khi các giá trị càng to thì các khối u có khả năng là ác tính. Theo logic đó, ta sẽ lấy các gía trị Min, Max cùng với biến mục tiêu để kiểm tra xem có đúng logic hay không, bằng 2 đoạn mã nguồn sau đây.

```
for column in X.columns:
    max_value = X[column].max()
    max_index = X[column].idxmax() # trả lại ví trị dòng có giá trị max
    corresponding_diagnosis = df.loc[max_index, 'diagnosis']
    print(f"Cột {column} có giá trị cao nhất = {max_value:4f}, Diagnosis:
{corresponding_diagnosis}")

for column in X.columns:
    min_value = X[column].min()
    min_index = X[column].idxmin() # trả lại ví trị dòng có giá trị min
    corresponding_diagnosis = df.loc[min_index, 'diagnosis']
    print(f"Cột  {column}  có  giá  trị  thấp  nhất  =  {min_value:4f},
Diagnosis: {corresponding_diagnosis}")
```

Kết quả:

```
Cột radius_mean có giá trị cao nhất = 28.110000, Diagnosis: 1
Cột texture_mean có giá trị cao nhất = 39.280000, Diagnosis: 1
Cột perimeter_mean có giá trị cao nhất = 188.500000, Diagnosis: 1
Cột area_mean có giá trị cao nhất = 2501.000000, Diagnosis: 1
Cột smoothness_mean có giá trị cao nhất = 0.163400, Diagnosis: 0
Cột compactness_mean có giá trị cao nhất = 0.345400, Diagnosis: 1
Cột concavity_mean có giá trị cao nhất = 0.426800, Diagnosis: 1
Cột concave points_mean có giá trị cao nhất = 0.201200, Diagnosis: 1
Cột symmetry_mean có giá trị cao nhất = 0.304000, Diagnosis: 1
Cột fractal_dimension_mean có giá trị cao nhất = 0.097440, Diagnosis: 1
Cột radius_se có giá trị cao nhất = 2.873000, Diagnosis: 1
Cột texture_se có giá trị cao nhất = 4.885000, Diagnosis: 0
Cột perimeter_se có giá trị cao nhất = 21.980000, Diagnosis: 1
Cột area_se có giá trị cao nhất = 542.200000, Diagnosis: 1
Cột smoothness_se có giá trị cao nhất = 0.031130, Diagnosis: 1
Cột compactness_se có giá trị cao nhất = 0.135400, Diagnosis: 1
Cột concavity_se có giá trị cao nhất = 0.396000, Diagnosis: 0
Cột concave points_se có giá trị cao nhất = 0.052790, Diagnosis: 0
Cột symmetry_se có giá trị cao nhất = 0.078950, Diagnosis: 1
Cột fractal_dimension_se có giá trị cao nhất = 0.029840, Diagnosis: 0
Cột radius_worst có giá trị cao nhất = 36.040000, Diagnosis: 1
Cột texture_worst có giá trị cao nhất = 49.540000, Diagnosis: 1
Cột perimeter_worst có giá trị cao nhất = 251.200000, Diagnosis: 1
Cột area_worst có giá trị cao nhất = 4254.000000, Diagnosis: 1
Cột smoothness_worst có giá trị cao nhất = 0.222600, Diagnosis: 1
Cột compactness_worst có giá trị cao nhất = 1.058000, Diagnosis: 1
Cột concavity_worst có giá trị cao nhất = 1.252000, Diagnosis: 0
Cột concave points_worst có giá trị cao nhất = 0.291000, Diagnosis: 1
Cột symmetry_worst có giá trị cao nhất = 0.663800, Diagnosis: 1
Cột fractal_dimension_worst có giá trị cao nhất = 0.207500, Diagnosis: 1
```

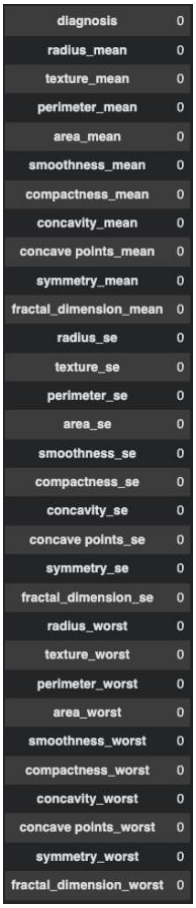
```
Cột radius_mean có giá trị thấp nhất = 6.981000, Diagnosis: 0
Cột texture_mean có giá trị thấp nhất = 9.710000, Diagnosis: 0
Cột perimeter_mean có giá trị thấp nhất = 43.790000, Diagnosis: 0
Cột area_mean có giá trị thấp nhất = 143.500000, Diagnosis: 0
Cột smoothness_mean có giá trị thấp nhất = 0.052630, Diagnosis: 0
Cột compactness_mean có giá trị thấp nhất = 0.019380, Diagnosis: 0
Cột concavity_mean có giá trị thấp nhất = 0.000000, Diagnosis: 0
Cột concave points_mean có giá trị thấp nhất = 0.000000, Diagnosis: 0
Cột symmetry_mean có giá trị thấp nhất = 0.106000, Diagnosis: 0
Cột fractal_dimension_mean có giá trị thấp nhất = 0.049960, Diagnosis: 1
Cột radius_se có giá trị thấp nhất = 0.111500, Diagnosis: 0
Cột texture_se có giá trị thấp nhất = 0.360200, Diagnosis: 0
Cột perimeter_se có giá trị thấp nhất = 0.757000, Diagnosis: 0
Cột area_se có giá trị thấp nhất = 6.802000, Diagnosis: 0
Cột smoothness_se có giá trị thấp nhất = 0.001713, Diagnosis: 0
Cột compactness_se có giá trị thấp nhất = 0.002252, Diagnosis: 0
Cột concavity_se có giá trị thấp nhất = 0.000000, Diagnosis: 0
Cột concave points_se có giá trị thấp nhất = 0.000000, Diagnosis: 0
Cột symmetry_se có giá trị thấp nhất = 0.007882, Diagnosis: 1
Cột fractal_dimension_se có giá trị thấp nhất = 0.000895, Diagnosis: 0
Cột radius_worst có giá trị thấp nhất = 7.930000, Diagnosis: 0
Cột texture_worst có giá trị thấp nhất = 12.020000, Diagnosis: 0
Cột perimeter_worst có giá trị thấp nhất = 50.410000, Diagnosis: 0
Cột area_worst có giá trị thấp nhất = 185.200000, Diagnosis: 0
Cột smoothness_worst có giá trị thấp nhất = 0.071170, Diagnosis: 0
Cột compactness_worst có giá trị thấp nhất = 0.027290, Diagnosis: 0
Cột concavity_worst có giá trị thấp nhất = 0.000000, Diagnosis: 0
Cột concave points_worst có giá trị thấp nhất = 0.000000, Diagnosis: 0
Cột symmetry_worst có giá trị thấp nhất = 0.156500, Diagnosis: 1
Cột fractal_dimension_worst có giá trị thấp nhất = 0.055040, Diagnosis: 1
```

Hình ảnh 2: Giá trị Max (bên trái) và giá trị Min (bên phải)

Trong kết quả ta thấy đa phần những giá trị Max, thường được gắn liền với những khối u ác tính và ngược lại giá trị Min, thường được gắn liền với nhưng khối u lành tính. Tuy nhiên có một số trường hợp ngoại lệ vs như giá trị Min, Max của biến Smoothness_mean (trung bình độ mịn) có giá trị bằng 0 – u lành tính. Việc quan sát các giá trị cực trị này giúp chúng ta nhận diện được ngưỡng phân tách sơ bộ giữa hai trạng thái bệnh lý. Tuy nhiên, sự tồn tại của các trường hợp ngoại lệ như biến smoothness_mean cho thấy các đặc trưng đơn lẻ đôi khi không đủ để đưa ra kết luận chính xác tuyệt đối.

- Điểm dự liệu bị khuyết (Null Value)

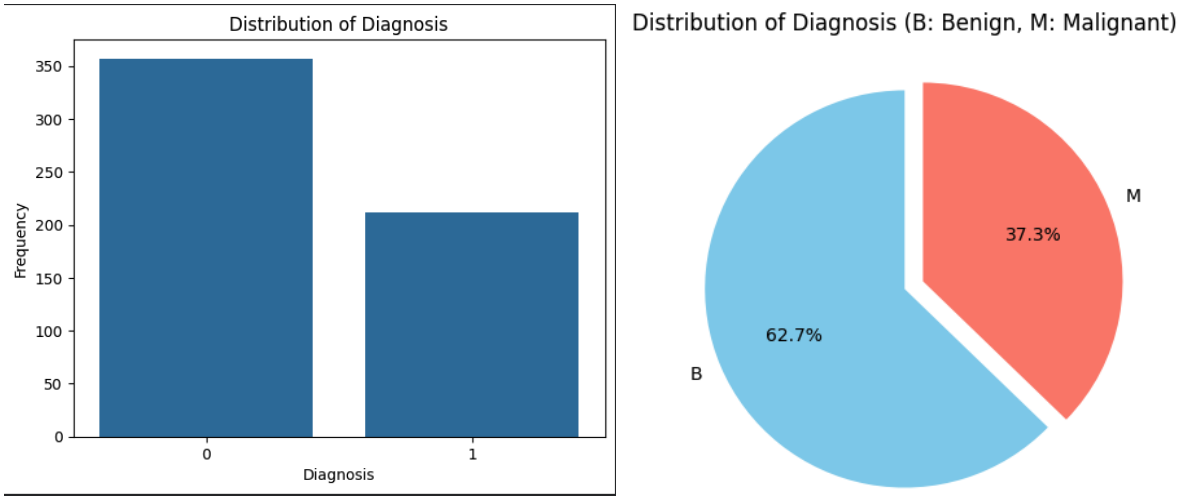
Bằng câu lệnh df.isnull().sum(), kết quả trả ra cho thấy tất cả các biến trong toàn bộ bộ dữ liệu không có giá trị nào bị khuyết.



Hình ảnh 3: Kết quả kiểm tra giá trị bị khuyết

2.2: Khám phá đặc trưng (Feature Exploration) và phân phối dữ liệu qua biểu đồ trực quan (Visualization)

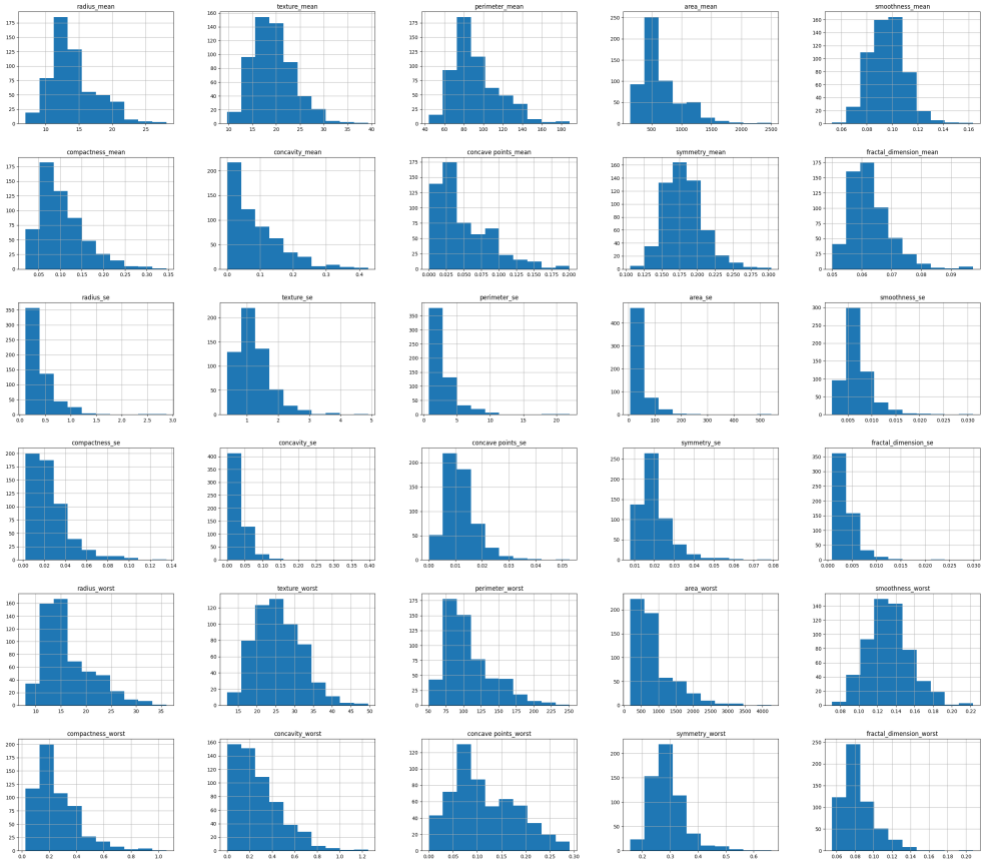
- Phân tích phân phối dữ liệu của biến mục tiêu



Hình ảnh 4&5: Phân phối của biến mục tiêu

Hai biểu đồ trên thể hiện sự phân bổ của biến mục tiêu trong tập dữ liệu chẩn đoán, được chia thành hai nhóm lành tính và ác tính. Biểu đồ cột bên trái cho thấy sự chênh lệch rõ rệt về số lượng mẫu, với nhóm lành tính chiếm ưu thế hơn hẳn khi đạt xấp xỉ 357 mẫu, trong khi nhóm ác tính chỉ có khoảng 212 mẫu. Tương ứng với số liệu đó, biểu đồ tròn bên phải minh họa tỷ lệ phần trăm cụ thể với nhóm lành tính chiếm 62,7% và nhóm ác tính chiếm 37,3% tổng thể dữ liệu. Sự phân bổ này cho thấy tập dữ liệu có sự mất cân bằng nhẹ (imbalance), một đặc điểm quan trọng cần lưu ý khi xây dựng các mô hình học máy để đảm bảo khả năng dự báo chính xác cho cả hai loại chẩn đoán, đặc biệt là đối với các trường hợp ác tính vốn có số lượng ít hơn.

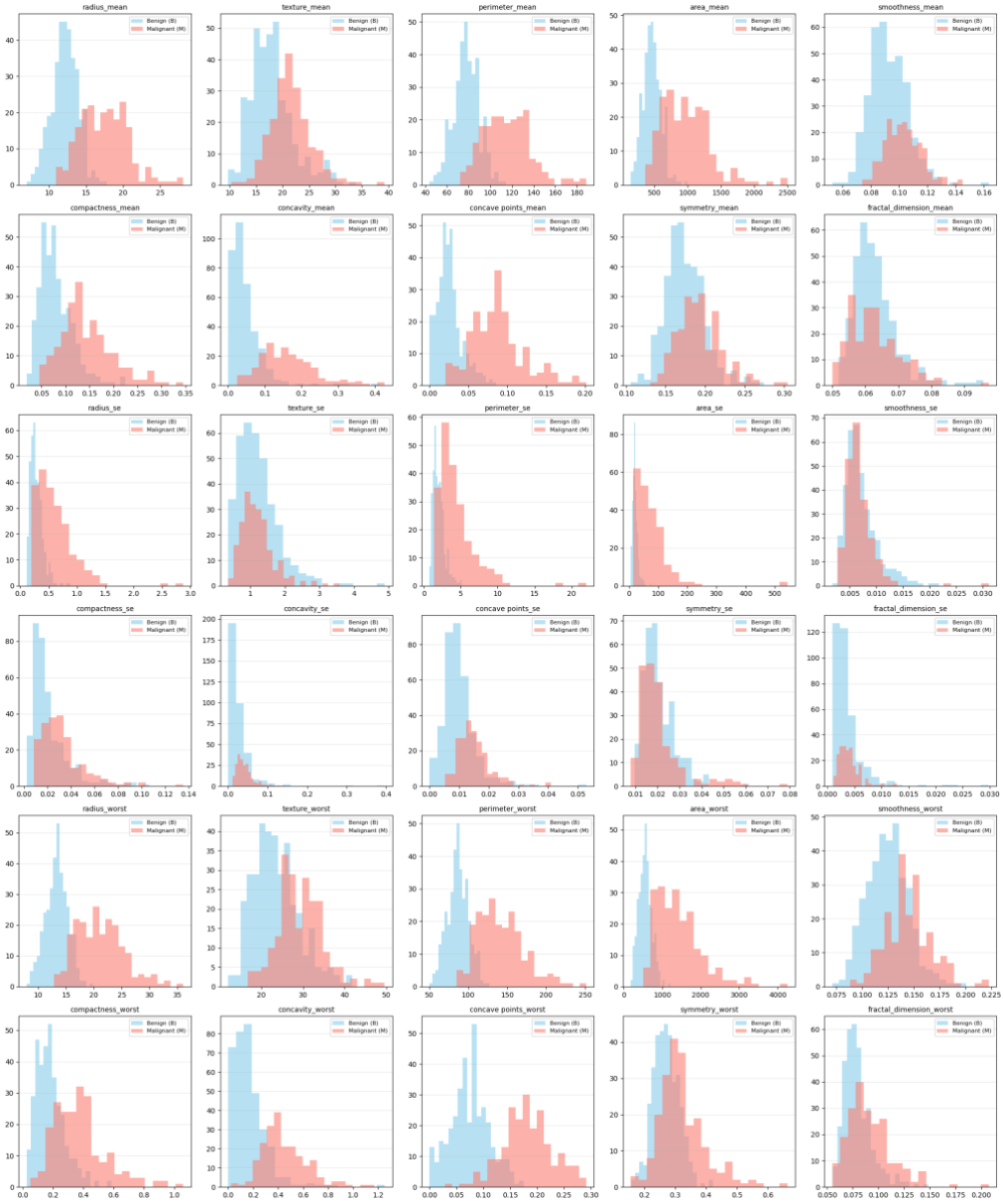
- Phân tích phân phối dữ liệu của tất cả các biến



Hình ảnh 6: Phân phối của tất cả các biến

Nhìn vào hệ thống biểu đồ phân phối của 30 biến số, ta có thể thấy một đặc điểm chung nổi bật là hầu hết các biến đều có dạng lệch phải (right-skewed). Điều này thể hiện rõ nhất ở các nhóm biến về kích thước như diện tích, chu vi, bán kính và các biến về độ phức tạp như concavity hay concave points. Việc các biểu đồ có đuôi kéo dài về phía bên phải cho thấy đa số các mẫu tế bào trong dataset tập trung ở các giá trị nhỏ (thường là khối u lành tính), nhưng có một số ít mẫu sở hữu giá trị cực lớn, tạo nên các đường cong lệch. Thông qua các biểu đồ này, chúng ta cũng có thể sơ bộ xác định sự hiện diện của các giá trị ngoại lai (outliers). Những cột đơn lẻ nằm tách biệt hẳn về phía bên phải của trục hoành chính là những điểm dữ liệu dị biệt, thường đại diện cho các trường hợp tế bào biến dị nghiêm trọng trong các khối u ác tính. Ngoài ra, đúng như tính chất sinh học đã bàn luận, toàn bộ các giá trị đo lường trên trục hoành đều lớn hơn 0, khẳng định tính hợp lệ của dữ liệu đầu vào vì các chỉ số vật lý của tế bào không thể mang giá trị âm.

- Sự khác biệt phân phối giữa khối u lành tính và ác tính

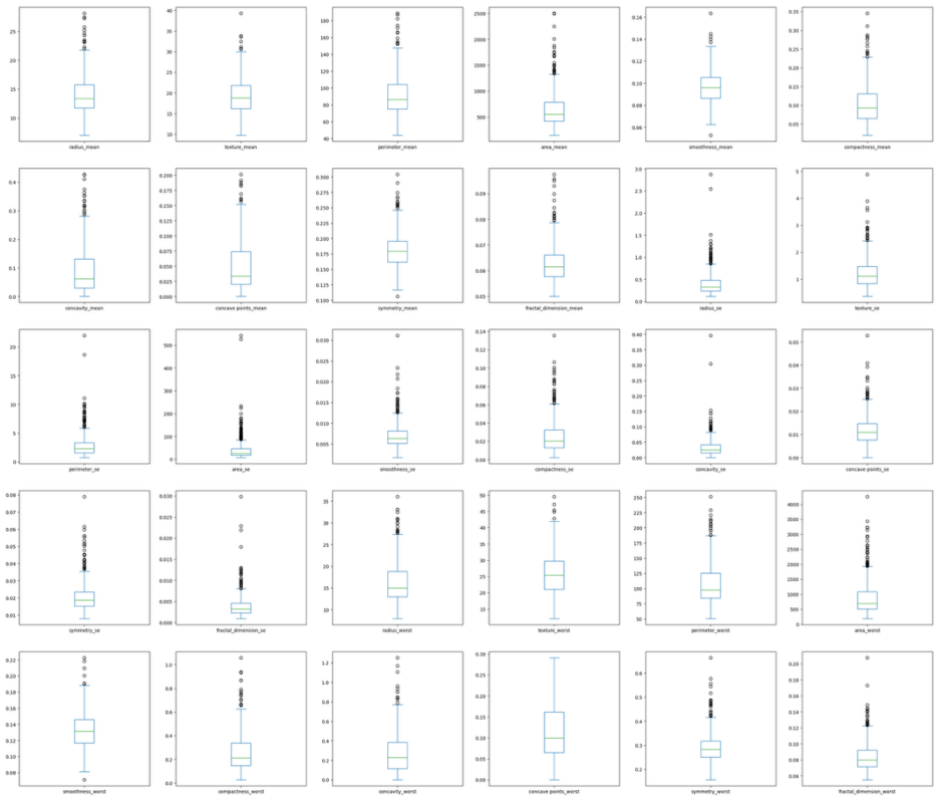


Hình ảnh 7: Phân phối của giữa khối u lành tính và ác tính

Các biến về kích thước và các biến mang giá trị "worst" tương ứng cho thấy sự tách biệt mạnh mẽ nhất. Trong khi nhóm lành tính (màu xanh) tập trung ở các giá trị nhỏ và thấp, thì nhóm ác tính (màu đỏ) lại phân bố rộng hơn và lệch hẳn về phía bên phải với các giá trị cao. Ngược lại, một số biến như kích thước Fractal hay độ mịn có sự chồng lấn rất lớn giữa hai màu, cho thấy các chỉ số này đơn lẻ rất khó để phân biệt loại khối u.

Về khoản tập trung nhiều điểm dữ liệu nhất, nhóm lành tính có mật độ tập trung cực kỳ cao và ổn định trong các khoảng giá trị thấp, ví dụ, biến diện tích trung bình của nhóm B chủ yếu nằm dưới mức 500, trong khi nhóm M thường bắt đầu từ mức 500 trở lên và trải dài đến hơn 2000. Điều này chứng tỏ tế bào lành tính có sự đồng nhất cao về kích thước, trong khi tế bào ác tính có xu hướng biến dị và phát triển không kiểm soát. Khi xét đến các giá trị ngoại lai (outliers), nhóm ác tính bộc lộ nhiều điểm dữ liệu dị biệt hơn ở phía đuôi bên phải, đặc biệt tại các biến như diện tích (worst) và chu vi (worst), trong khi nhóm lành tính dù ít ngoại lai hơn nhưng các điểm này thường nằm sát ngưỡng biên giới của nhóm ác tính, dễ gây ra sự chồng lấn trong phân loại sơ bộ. Nên có thể nói các tế bào ác tính thường đạt tới những con số cực đại, phản ánh mức độ biến dị và tính chất xâm lấn đặc trưng của ung thư so với sự ổn định về mặt cấu trúc của các khối u lành tính.

- Dùng biểu đồ Box plot phân tích phân tích giá trị ngoại lai

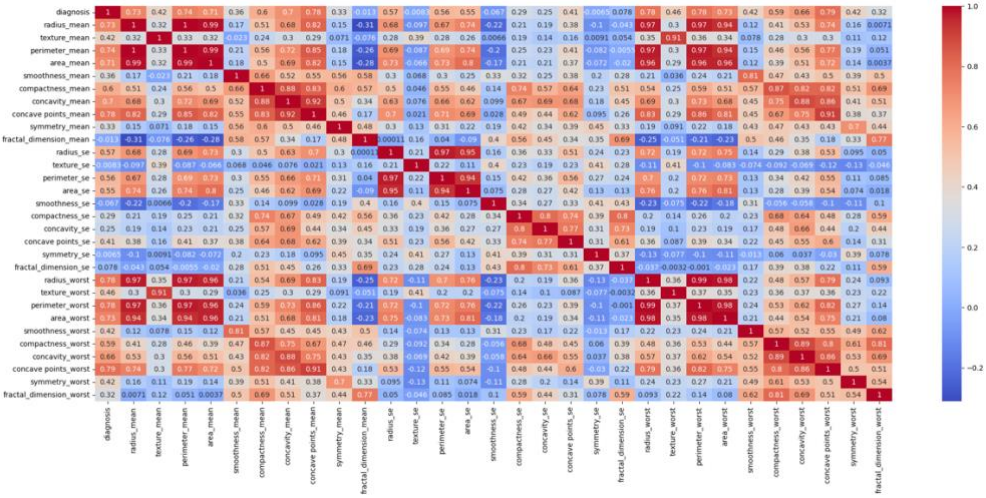


Hình ảnh 8: Biểu đồ Boxplot của tất cả các biến

Quan sát hệ thống 30 biểu đồ Box Plot, chúng ta có thể nhận thấy một đặc điểm thống kê cực kỳ quan trọng là có tới 29 trên tổng số 30 biến xuất hiện các giá trị ngoại lai, biểu thị bằng các

điểm chấm rời rạc nằm phía trên ranh giới của râu biểu đồ (upper whiskers). Việc các giá trị ngoại lai xuất hiện dày đặc và hầu như ở mọi chỉ số cho thấy sự biến thiên rất lớn trong hình thái tế bào giữa các mẫu bệnh phẩm. Về mặt y sinh, những "điểm dị biệt" này thường mang thông tin quý giá vì chúng đại diện cho các tế bào biến dị cực đoan trong các khối u ác tính. Tuy nhiên, về mặt kỹ thuật máy học, sự tồn tại của quá nhiều outliers có thể gây nhiễu và làm sai lệch đáng kể tới giá trị trung bình cũng như độ lệch chuẩn, dẫn đến việc các mô hình nhạy cảm với khoảng cách hoặc các thuật toán dựa Logistic Regression dễ bị mất phương hướng khi tìm kiếm ranh giới phân loại tối ưu (Ansari et al. 2024).

- Ma trận tương quan



Hình ảnh 9: Ma trận tương quan

Khi xét đến tương quan với biến mục tiêu trong ma trận tương quan, ta thấy một nhóm các đặc trưng có hệ số tương quan dương rất mạnh (hầu hết trên 0.7). Nổi bật nhất là số điểm lõm (worst) với giá trị 0.78, chu vi (worst) có giá trị 0.78 , số điểm lõm (mean) với giá trị 0.78. Điều này cho thấy các chỉ số về độ lõm và kích thước tế bào ở giai đoạn tệ nhất là những tín hiệu quan trọng nhất để phân loại khối u ác tính. Ngược lại, các biến như kích thước fractal (-0.012), kết cấu (SE) với giá trị (-0.0083) và độ min (SE) có giá trị (-0.067) hầu như không có mối tương quan tuyến tính với kết quả chẩn đoán, gợi ý rằng chúng có thể không đóng góp nhiều giá trị trong các mô hình tuyến tính đơn giản.

Một điểm cực kỳ quan trọng cần lưu ý là hiện tượng đa cộng tuyến (Multicollinearity) giữa các biến độc lập. Nói qua về hiện tượng đa cộng tuyến xảy ra khi các biến độc lập có mối quan hệ phụ thuộc lẫn nhau quá chặt chẽ, điều này không chỉ gây ra sự dư thừa thông tin mà còn làm cho các hệ số của mô hình máy học trở nên không ổn định, dễ bị nhiễu và gây khó khăn trong việc xác định chính xác đâu mới là yếu tố thực sự quyết định đến kết quả chẩn đoán (Alin 2010). Các nhóm biến như bán kính, chu vi và diện tích (ở cả ba dạng: mean, se, worst) có hệ số tương quan gần như tuyệt đối, dao động từ 0.94 đến 0.99. Điều này hoàn toàn dễ hiểu về mặt hình học vì chu vi và diện tích đều được tính toán dựa trên bán kính. Việc giữ lại tất cả các biến này trong mô hình có thể dẫn đến dư thừa thông tin, làm nhiễu mô hình và gây khó khăn

cho việc giải thích tầm quan trọng của từng đặc trưng. Ngoài ra, các đặc trưng về hình dáng như độ nén, độ lõm và số điểm lõm cũng có mối gắn kết chặt chẽ với nhau (> 0.8). Phân tích này là cơ sở then chốt để chúng ta có thể thực hiện những kỹ thuật khác, nhằm loại bỏ các biến dư thừa và tối ưu hóa hiệu suất cho mô hình máy học ở các bước tiếp theo.

III: Tiền sử lý dữ liệu - Data Preprocessing

3.1: Chia bộ dữ liệu ra Train và Test

Trong bước này, chúng ta tiến hành chia tập dữ liệu thành hai phần riêng biệt là tập huấn luyện (training set) và tập kiểm tra (testing set) với tỷ lệ 70/30, nghĩa là 70% dữ liệu sẽ được dùng để máy học rèn luyện và 30% còn lại được giữ lại để đánh giá khách quan năng lực của mô hình trên dữ liệu mới. Việc thiết lập tham số "shuffle=True" là vô cùng cần thiết để xáo trộn ngẫu nhiên thứ tự các bản ghi trước khi chia, giúp loại bỏ các yếu tố thiên kiến nếu dữ liệu gốc vô tình được sắp xếp theo một quy luật nào đó, đảm bảo mô hình không học theo trình tự nhập liệu mà học theo đặc trưng thực tế. Đặc biệt, tham số "stratify=y" đóng vai trò then chốt trong việc duy trì sự cân bằng của các nhóm nhãn; nó đảm bảo rằng tỷ lệ giữa hai lớp lành tính và ác tính trong cả hai tập train và test đều đồng nhất với tỷ lệ gốc của bộ dữ liệu ban đầu. Điều này giúp ngăn ngừa tình trạng tập kiểm tra bị lệch nhãn (ví dụ như chỉ toàn ca lành tính), từ đó giúp kết quả đánh giá mô hình trở nên tin cậy và phản ánh chính xác khả năng chẩn đoán trong thực tế y khoa.

3.2: Xử lý dữ liệu ngoại lai

```
def handle_outliers_train_test(X_train, X_test):

    X_train_clean = X_train.copy()
    X_test_clean = X_test.copy()
    outlier_bounds = {}

    for column in X_train.columns:
        Q1 = X_train[column].quantile(0.25)
        Q3 = X_train[column].quantile(0.75)
        IQR = Q3 - Q1
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR

        outlier_bounds[column] = {'lower': float(lower_bound),'upper':
float(upper_bound)}

        # Đếm outliers
        outliers_train = ((X_train[column] < lower_bound) |
                          (X_train[column] > upper_bound)).sum()
        outliers_test = ((X_test[column] < lower_bound) |
                         (X_test[column] > upper_bound)).sum()

        X_train_clean[column] = X_train[column].clip(lower_bound,
upper_bound)
        X_test_clean[column] = X_test[column].clip(lower_bound,
upper_bound)
```

```
        # In kết quả nếu có outliers
        if outliers_train > 0 or outliers_test > 0:
            print(f"{column}: Train outliers={outliers_train}, "
                  f"Test outliers={outliers_test}")


    return X_train_clean, X_test_clean, outlier_bounds

X_train, X_test, outlier_bounds = handle_outliers_train_test(X_train,
X_test)
```

Quy trình xử lý giá trị ngoại lai được thực hiện thông qua phương pháp khoảng biến thiên phân vị (IQR), một kỹ thuật giúp xác định ranh giới hợp lệ của dữ liệu mà không phụ thuộc vào giả định phân phối chuẩn. Với kết quả thực tế cho thấy có tới 29 trên tổng số 30 biến xuất hiện điểm dị biệt, việc xử lý là bắt buộc để ổn định mô hình,tuy nhiên, thay vì xóa bỏ các hàng dữ liệu này, chúng ta sử dụng kỹ thuật Capping (Winsorizing).Kỹ thuật nêu trên áp dụng trong việc xử lý dữ liệu ngoại lai bằng cách thay vì xoá dữ liệu, ta thay những dữ liệu cực đoan bằng những điểm dữ liệu ít cực đoan hơn và trong trường hợp này ta dùng giá trị Min và Max (Wilcox 2003). Quyết định này xuất phát từ đặc thù của bộ dữ liệu ung thư vú có quy mô mẫu khá nhỏ (chỉ 567 bản ghi) và mỗi điểm dữ liệu y tế đều cực kỳ quý giá, chứa đựng những thông tin sinh học quan trọng mà nếu xóa đi sẽ làm mất tính đại diện của tập mẫu. Bằng cách quy đổi các giá trị vượt ngưỡng về mức tối thiểu (lower bound) hoặc tối đa (upper bound), chúng ta vừa giảm thiểu được tác động nhiễu của các giá trị cực đoan, vừa giữ lại được toàn bộ 567 mẫu cho quá trình huấn luyện.

Đặc biệt, để đảm bảo tính nghiêm ngặt trong kiểm thử máy học và tránh hiện tượng Rò rỉ dữ liệu (Data Leakage) (IBM 2024), các ngưỡng thống kê như Q1, Q3 và IQR chỉ được tính toán dựa trên tập huấn luyện (X_train). Việc fit các ngưỡng này từ tập huấn luyện rồi mới áp dụng (transform) sang tập kiểm tra (X_test) đảm bảo rằng mô hình không được tiếp cận bất kỳ thông tin thống kê nào từ dữ liệu kiểm tra trước khi dự đoán.

3.3: . Áp dụng kỹ thuật chuẩn hoá StandardScalar

```
# Scale các biến
scaler = StandardScaler()
X_train_scaled= scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Trong bước tiền xử lý dữ liệu, kỹ thuật chuẩn hóa bằng StandardScaler được lựa chọn để đưa các đặc trưng về cùng một quy mô đơn vị (scale), đảm bảo phân phối của dữ liệu có giá trị trung bình bằng 0 và độ lệch chuẩn bằng 1 (scikit-learn 2019). StandardScaler giúp bảo toàn hình dạng phân phối của dữ liệu gốc và đảm bảo rằng các thuộc tính có tầm giá trị lớn không chiếm ưu thế một cách thiếu khách quan so với các thuộc tính có tầm giá trị nhỏ. Việc đồng bộ hóa thang đo là điều kiện tiên quyết đối với các thuật toán dựa trên tính toán khoảng cách giúp quá trình hội tụ diễn ra nhanh và ổn định hơn. Đặc biệt, quy trình thực hiện được kiểm soát chặt chẽ bằng cách áp dụng fit_transform trên tập huấn luyện (Train set) và chỉ transform trên

tập kiểm tra (Test set). Phương pháp này giúp ngăn ngừa hiện tượng rò rỉ dữ liệu (data leakage), đảm bảo rằng các tham số thống kê của tập kiểm tra hoàn toàn không được biết trước trong quá trình huấn luyện.

IV: Huấn luyện mô hình - Model training

4.1: Các kỹ thuật được áp dụng

4.1.1: Cross-validation

Để đảm bảo mô hình có khả năng tổng quát hóa tốt và đánh giá hiệu suất một cách khắt khe nhất, chúng ta áp dụng kỹ thuật Stratified KFold Cross-Validation với tham số CV = 5. Trong bối cảnh bộ dữ liệu y tế thường gặp tình trạng mất cân bằng nhãn, việc nên sử dụng biến thể "Stratified" là hợp lý vì nó đảm bảo rằng trong mỗi lần chia nhỏ (fold), tỷ lệ giữa hai lớp lành tính và ác tính luôn được duy trì ổn định và đồng nhất với cấu trúc của tập dữ liệu gốc (Song 2025). Với cấu hình này, tập dữ liệu được chia thành 5 phần, mô hình sẽ trải qua 5 lần lặp, mỗi lần huấn luyện trên 4 phần và kiểm thử trên phần còn lại để thu được 5 giá trị F1-score độc lập. Thay vì chỉ sử dụng Accuracy (độ chính xác) vốn dễ bị đánh lừa bởi lớp đa số, chúng ta ưu tiên F1-score vì đây là giá trị trung bình điều hòa giữa Precision và Recall, giúp đánh giá chính xác khả năng nhận diện các ca bệnh thực tế mà không bỏ sót các ca ác tính (Lee 2025). Kết quả F1 tổng thể (overall F1-score) sẽ được tính bằng cách lấy trung bình cộng của 5 lần thử nghiệm này, giúp loại bỏ các sai số ngẫu nhiên và cung cấp một thước đo tin cậy, phản ánh đúng năng lực chẩn đoán bền vững của mô hình trên các tập dữ liệu chưa từng tiếp cận.

4.1.2: Điều chỉnh siêu tham số (Tuning Hyperparameter)

Trong thuật toán, việc thực hiện tuning hyperparameter (tinh chỉnh siêu tham số) là một bước tối ưu hóa then chốt nhằm tinh chỉnh ranh giới quyết định giữa hai nhóm nhãn lành tính và ác tính, giúp mô hình đạt đến trạng thái cân bằng lý tưởng nhất. Khác với các trọng số mà mô hình tự học, các siêu tham số như Regularization Strength (C) hay các loại Penalty (L1, L2) cần được chúng ta thiết lập thủ công để kiểm soát mức độ khắt khe của mô hình đối với các biến số phức tạp. Lợi ích lớn nhất của quy trình này là khả năng ngăn chặn hiện tượng overfitting bằng cách điều chỉnh tham số C, khi giá trị C nhỏ sẽ tăng cường tính chính quy hóa để đơn giản hóa mô hình, trong khi C lớn cho phép mô hình bám sát dữ liệu huấn luyện hơn. Thông qua việc tìm kiếm tổ hợp tối ưu kết hợp với kỹ thuật cross validation, mô hình có thể cải thiện đáng kể chỉ số F1-score mà còn trở nên bền bỉ hơn trước các nhiễu từ giá trị ngoại lai, đảm bảo rằng xác suất dự đoán đưa ra có độ tin cậy cao nhất trong bối cảnh chẩn đoán y khoa.

4.1.3: Chỉ số đánh giá (Metrics)

Ở trên F1-score được nhắc đến khá nhiều, nhưng một trong những lý do trong bài báo cáo này F1-score được làm báo cáo chính là vì trong bộ dữ liệu này của dữ liệu y tế thường xuyên đối mặt với sự mất cân bằng giữa số lượng ca lành tính và ác tính, khiến chỉ số Accuracy (độ chính xác) trở nên dễ gây hiểu lầm. Thế nên, F1-score đóng vai trò là cầu nối giúp cân bằng giữa

Precision (độ tin cậy của các ca được dự đoán là ác tính) và Recall (khả năng không bỏ sót bất kỳ ca ác tính thực sự nào), từ đó phản ánh sát thực nhất hiệu quả chẩn đoán lâm sàng.

4.2: Giới thiệu các mô hình học máy

4.2.1: Hồi quy Logistic (Logistic Regression)

Hồi quy Logistic là 1 mô hình được dùng cho bài toán phân loại nhị phân với biến mục tiêu chỉ nhận 2 giá trị (lành tính và ác tính). Hồi quy Logistic sử dụng hàm Sigmoid để nén đầu ra của một phương trình tuyến tính vào khoảng giá trị từ 0 đến 1. Giá trị đầu ra này được hiểu là xác suất để một mẫu dữ liệu thuộc về một lớp cụ thể. Thông thường, một ngưỡng (threshold) mặc định là 0.5 (có thể điều chỉnh) sẽ được thiết lập với nếu xác suất lớn hơn 0.5, mẫu đó được phân loại là lớp 1 (ác tính), ngược lại là lớp 0 (lành tính). Bằng cách phân loại nêu trên, mô hình hồi quy Logistic, mô hình có thể giúp bác sĩ dự đoán các khối u và giúp người bệnh được chữa trị 1 cách sớm nhất.

4.2.2: Cây quyết định (Decision Tree)

Cây quyết định là một thuật toán học máy giám sát mạnh mẽ với cấu trúc phân cấp tương tự như một sơ đồ luồng, giúp đưa ra quyết định dựa trên các quy tắc "if-then" được học từ dữ liệu (MLJourney 2025). Cấu trúc của cây bắt đầu bằng một nút gốc (Root Node), nơi chứa đặc trưng có khả năng phân loại dữ liệu tốt nhất. Từ đó, cây được chia nhánh thành các nút trung gian (Internal Nodes) đại diện cho các điều kiện kiểm tra thuộc tính, và cuối cùng kết thúc tại các nút lá (Leaf Nodes), nơi đưa ra kết quả phân loại cuối cùng (như lành tính hoặc ác tính) kèm theo điều kiện nào đó, ví dụ như biến A < 10. Quá trình hình thành các nút và chia nhánh được thực hiện thông qua việc đo lường độ tinh khiết của dữ liệu. Hai tiêu chí phổ biến nhất để lựa chọn đặc trưng chia nhánh là chỉ số Gini (Gini Impurity) và độ lợi thông tin (Information Gain) dựa trên Entropy. Trong khi Gini đo lường xác suất một phần tử bị phân loại sai, thì Entropy đo lường mức độ hỗn loạn của thông tin, mục tiêu của mô hình là tìm cách chia sao cho các nút con sau khi tách có độ tinh khiết cao nhất (giá trị Gini hoặc Entropy thấp nhất).

4.3: Huấn luyện mô hình

```
def validate_model_kfold(model, param_dist, X_input, y_input, n_splits,
n_iter):
    skf = StratifiedKFold(n_splits=n_splits, shuffle=True,
random_state=42)

    search = RandomizedSearchCV(
        estimator=model,
        param_distributions=param_dist,
        n_iter=n_iter,
        cv=skf,
        scoring='f1',
        n_jobs=-1,
        verbose=1,
```

```
        random_state = 42
    )

    search.fit(X_input, y_input)

    # Lấy index của best parameters
    best_index = search.best_index_
    print("\n")
    print("Chi tiết từng fold với bộ siêu tham số tốt nhất:")


    # Lấy accuracy từng fold từ cv_results_
    fold_scores = []
    for fold in range(n_splits):
        fold_score =
search.cv_results_[f'split{fold}_test_score'][best_index]
        fold_scores.append(fold_score)
        print(f"Fold {fold + 1}: f1 = {fold_score * 100:.2f}%")

    print("\n")
    print(f"Bộ siêu tham số tốt nhất: {search.best_params_}")

    # Fit best model với toàn bộ dữ liệu
    best_model = search.best_estimator_

    # Test model
    y_pred = best_model.predict(X_test_scaled)

    f1 = f1_score(y_test, y_pred)
    report = classification_report(y_test, y_pred)
    conf_matrix = confusion_matrix(y_test, y_pred)

    # In kết quả
    print(f"\nCV f1 score: {search.best_score_:.2%}")
    print(f"Test f1 score: {f1 * 100:.2f}%")
    print("Classification Report:\n", report)
    print("Confusion Matrix:\n", conf_matrix)

    return best_model
```

Thay vì chỉ kiểm tra một bộ tham số cố định, ta tạo ra function validate_model_kfold sử dụng RandomizedSearchCV để tìm kiếm thông minh trong không gian các siêu tham số, giúp tiết kiệm thời gian tính toán mà vẫn đạt hiệu quả cao so với GridSearchCV. Theo 1 bài nghiên cứu về áp dùng mô hình học máy trong dữ liệu y khoa có tên " A Comparative Study of Machine Learning Models for Heart Disease Prediction Using Grid Search and Random Search for Hyperparameter Tuning " chỉ ra dành việc sử dụng Random Search sẽ tiết kiệm chi phí hơn tại nó sẽ check ngẫu nhiên tập hợp của các tham số và trong đa số những trường hợp thường tìm được bộ tham số tối ưu nhanh hơn Grid Search, bởi phương pháp này tìm kiếm theo dạng lưới và nó sẽ kiểm tra từng bộ tham số (Arshad et al. 2023).

Việc tích hợp StratifiedKFold như đã nói ở trên (CV=5) đảm bảo rằng trong mỗi lần thử nghiệm, tỷ lệ các lớp dữ liệu luôn được bảo toàn, đặc biệt quan trọng khi dữ liệu có sự chênh lệch giữa các ca B và M. Đoạn code không chỉ dừng lại ở việc tìm ra bộ tham số tốt nhất mà còn in chi tiết điểm số F1 của từng fold để kiểm tra độ ổn định của mô hình. Điểm hay nhất của quy trình này là tính minh bạch và nghiêm ngặt bởi sau khi "fit" trên toàn bộ dữ liệu huấn luyện với những tham số tối ưu nhất, mô hình (best_estimator_) mới được mang đi đánh giá cuối cùng trên tập kiểm tra độc lập (X_test_scaled). Việc xuất ra đồng thời Classification Report và Confusion Matrix trên tập test cung cấp một cái nhìn đa chiều về khả năng phân loại, giúp chúng ta không chỉ biết được độ chính xác tổng thể mà còn hiểu rõ mô hình có đang bỏ sót ca bệnh (False Negative) nào hay không.

V: Phân tích và đáng giá mô hình - Model analysis & evaluation

5.1: Hồi quy Logistic (Logistic Regression)

```
param_dist_lr = {
    'penalty': ['l1','l2'],
    'solver': ['liblinear'],
    'C': [0.001,0.01,0.1],
    'class_weight':["balanced"]
}

logistic_regression_model= validate_model_kfold(
    LogisticRegression(random_state = 42, max_iter= 1000),
    param_dist_lr,
    X_train_scaled,
    y_train,
    n_splits = 5,
    n_iter = 20)
```

Kết quả từ quy trình tối ưu hóa cho thấy mô hình Logistic Regression đạt được hiệu suất tốt và ổn định. Với bộ siêu tham số tốt nhất gồm 'C': 0.1, 'penalty': 'l2' và đặc biệt là 'class_weight': 'balanced', mô hình đã giải quyết hiệu quả vấn đề mất cân bằng dữ liệu, đẩy điểm CV f1-score trung bình lên mức 95.96%. Sự ổn định này được minh chứng qua kết quả của 5 fold, với các giá trị F1 dao động ở mức cao từ 91.80% đến 98.25%, cho thấy mô hình không bị phụ thuộc vào một cách chia dữ liệu cụ thể nào. Trên tập kiểm tra độc lập (Test set), mô hình đạt Test f1-score lên tới 99.21% (với seed = 42), một con số gần như lý tưởng cho bài toán chẩn đoán y khoa. Quan sát Ma trận nhầm lẫn (Confusion Matrix), chúng ta thấy mô hình đã phân loại chính xác tuyệt đối 107 ca lành tính (lớp 0) và chỉ bỏ sót duy nhất 1 ca ác tính (lớp 1) trong tổng số 64 ca. Chỉ số Recall cho lớp 1 đạt 0.98, khẳng định khả năng nhận diện bệnh nhân ung thư cực tốt, hạn chế tối đa nguy cơ bỏ lọt bệnh, yếu tố quan trọng nhất trong các hệ thống hỗ trợ quyết định y học. Bên cạnh đó khi chia tập test với những seed khác thì mô hình có độ chính xác rơi vào khoảng ~96-98% chứng minh sự ổn định của mô hình.

Áp dụng L2 (Ridge Regression) vào mô hình ở đây hợp lý bởi, thay vì loại bỏ các biến bằng cách hép các hệ số này bằng 0, L2 thêm một khoản phạt bằng bình phương độ lớn các hệ số vào hàm mất mát, giúp đẩy các hệ số này về gần 0 nhưng không triệt tiêu chúng (IBM 2023).

Điều nay giúp giữu lại tôi đa thông tin bởi dữ liệu về y khoa đặc biệt là ung thư rất quý giá và nếu cân nhắc đến yếu tố bên ngoài, dữ liệu y khoa có tính chất phức tạp nên nếu không phải nếu không phải là một chuyên gia y khoa dày dạn kinh nghiệm, người làm dữ liệu rất khó có thể tự tin quyết định loại bỏ bất kỳ biến nào mà không sợ làm mất đi những tín hiệu dự báo quan trọng, dù cho biến đó có vẻ không gây ảnh hưởng trực tiếp lên mô hình tại thời điểm phân tích.

5.2: Cây quyết định (Decision Tree)

```
param_dist_dt = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [None, 3,5,7],
    'min_samples_split': [3,5,7,10,20],
    'min_samples_leaf': [1,2,3,4,8]
 }

# Test lại bộ hyperparameters tốt nhất để xem kết quả từng fold
decision_tree = validate_model_kfold(
    DecisionTreeClassifier(random_state = 42),
    param_dist_dt,
    X_train_scaled,
    y_train,
    n_splits = 5,
    n_iter = 20)
```

Kết quả thực nghiệm cho thấy mô hình Decision Tree đạt hiệu suất khá tốt sau khi được tối ưu hóa. Thông qua quá trình tìm kiếm với 100 lần thử nghiệm, bộ siêu tham số lý tưởng nhất được xác định bao gồm criterion='entropy', max_depth=5, min_samples_split=3 và min_samples_leaf=4. Việc giới hạn độ sâu của cây ở mức 5 và thiết lập số lượng mẫu tối thiểu tại các nút lá giúp mô hình kiểm soát tốt hiện tượng Overfitting, đảm bảo cây không mọc quá sâu vào các chi tiết gây nhiễu. Điểm số CV f1-score trung bình đạt 90.37%, mặc dù có sự biến động giữa các fold (từ 85.71% đến 94.92%) có thể bản chất phân tách dữ liệu của cây quyết định thường nhạy cảm hơn so với hồi quy.

Trên tập kiểm tra, mô hình đạt Test f1-score là 91.67%. Quan sát Ma trận nhầm lẫn (Confusion Matrix), mô hình phân loại chính xác gần như tuyệt đối các ca lành tính (106/107 ca). Tuy nhiên, đối với các ca ác tính (lớp 1), mô hình vẫn bỏ sót 9 ca (False Negative) và đạt chỉ số Recall là 0.86. Điều này cho thấy mặc dù Decision Tree mang lại dễ hiểu về mặt quy tắc logic, nhưng trong bài toán y khoa này, nó vẫn có xác suất bỏ lọt bệnh cao hơn so với mô hình hồi quy Logistic đã phân tích trước đó mà bỏ lỡ FN có thể gây ra việc người bệnh không được chữa trị kịp thời và có thể nguy hiểm đến tương lai của người bệnh.

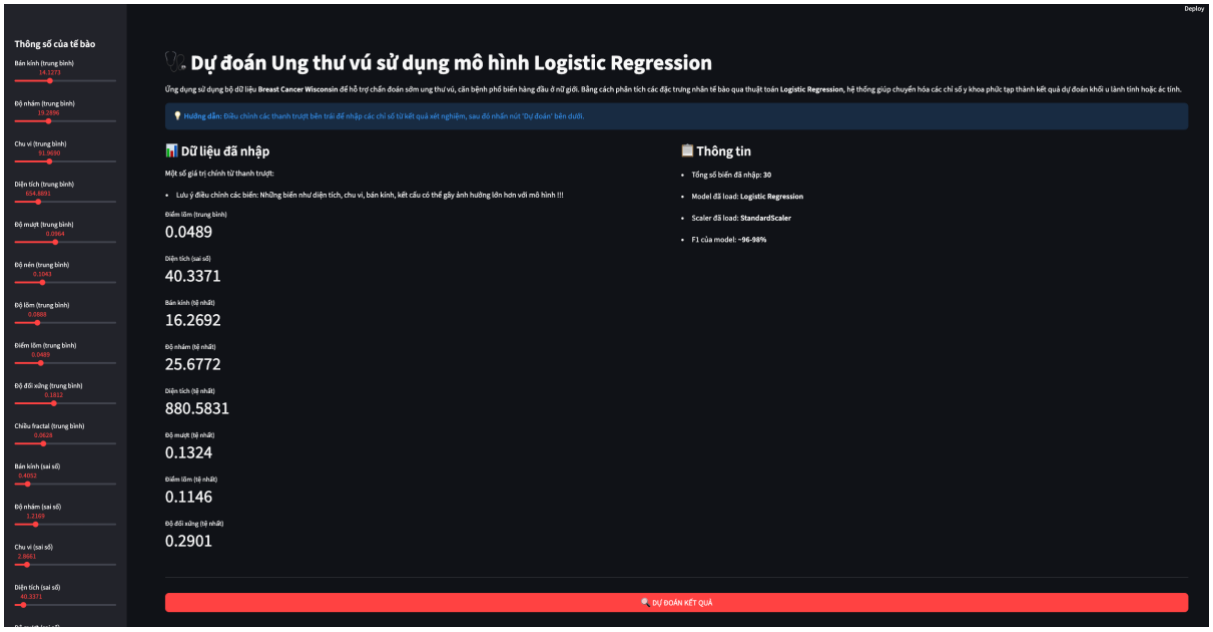
VI: Streamlit

Sau khi tiến hành so sánh đối chiếu hiệu suất giữa các thuật toán, tôi quyết định lựa chọn mô hình Logistic Regression làm mô hình cuối cùng để dự đoán bản chất khối u. Lý do chính nằm ở sự vượt trội về chỉ số F1-score (99.21%) và khả năng tối ưu hóa cực tốt của nó trước hiện

tượng đa cộng tuyến nhờ cơ chế L2 Regularization. So với Decision Tree, Logistic Regression không chỉ đạt độ chính xác cao hơn mà còn có độ ổn định (stability) tốt hơn giữa các fold dữ liệu, đồng thời giảm thiểu tối đa các ca âm tính giả (chỉ bỏ sót 1 ca so với 9 ca ở Decision Tree) đảm bảo an toàn trong chẩn đoán ung thư.

Để đưa nghiên cứu này vào thực tiễn, mô hình sẽ được triển khai trên nền tảng Streamlit. Đây là một framework mã nguồn mở mạnh mẽ bằng ngôn ngữ Python, cho phép biến các script phân tích dữ liệu thành các ứng dụng web tương tác một cách nhanh chóng và trực quan. Với Streamlit, người dùng (như các bác sĩ hoặc kỹ thuật viên) có thể dễ dàng nhập bằng thanh trượt các thông số hình thái tế bào trực tiếp trên giao diện web và nhận kết quả dự đoán ngay lập tức từ mô hình Logistic Regression đã được huấn luyện. Có một số lưu ý nhỏ như bên trong web đã đề cập, như là có một số biến sẽ gây ảnh hưởng nhiều hơn đến kết quả dự đoán, nên người dùng cần lưu ý điều chỉnh một cách chính xác nhất. Cách để chạy web, người dùng cần "run all" trong file ML09_gg_collab rồi 4 file pkl sẽ được lưu lại. Xong đó người dùng cần để 4 file pkl và file streamlit_application.py trong 1 tập và chạy

Thông qua streamlit, ta thấy được sự kết hợp giữa một thuật toán có độ tin cậy cao và một giao diện thân thiện giúp rút ngắn khoảng cách từ lý thuyết học máy đến ứng dụng lâm sàng thực tế.



Hình ảnh 10: Giao diện web dự đoán khối u

Kết luận

Việc chẩn đoán sớm và chính xác đóng vai trò sống còn đối với khả năng sinh tồn của bệnh nhân ung thư vú. Qua nghiên cứu, mô hình Logistic Regression kết hợp L2 Regularization đã chứng minh hiệu quả tốt trong việc xử lý dữ liệu y khoa phức tạp và đa cộng tuyến, giúp giữ lại tối đa thông tin quý giá mà không cần can thiệp thủ công từ người không có chuyên môn.

Bằng việc ưu tiên chỉ số F1-score, mô hình đạt được sự cân bằng tối ưu giữa Precision và Recall, từ đó giảm thiểu nguy cơ âm tính giả để tránh bỏ lọt bệnh nhân trong y tế. Cuối cùng, việc triển khai trên nền tảng Streamlit đã biến mô hình thành một công cụ hỗ trợ lâm sàng trực quan, cho phép dự đoán tức thì và hỗ trợ các bác sĩ đưa ra quyết định điều trị chính xác, kịp thời.

Reference

Alin A (2010) 'Multicollinearity', *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(3):370–374, doi:https://doi.org/10.1002/wics.84.

Ansari S, Nassif AB, Mahmoud S, Sohaib Majzoub, Eqab Almajali, Anwar Jarndal, Bonny T, Alnajjar KA and Hussain A (2024) 'Impact of Outliers on Regression and Classification Models: An Empirical Analysis', *IEEE*211–218, doi:https://doi.org/10.1109/dese63988.2024.10912020.

Arshad S, Syed, Ali M, Hashmi MU, Manan A and Ahmad A (2023) 'A Comparative Study of Machine Learning Models for Heart Disease Prediction Using Grid Search and Random Search for Hyperparameter Tuning', *Journal of Computing & Biomedical Informatics*, 8(01), https://jcbi.org/index.php/Main/article/view/697, accessed 11 January 2026.

IBM (2023) *Ridge Regression*, *Ibm.com*, https://www.ibm.com/think/topics/ridge-regression, accessed 11 January 2026.

IBM (2024) *Data leakage machine learning*, *Ibm.com*, https://www.ibm.com/think/topics/data-leakage-machine-learning, accessed 10 January 2026.

Lee S (2025) *The Ultimate F1 Score Guide for Imbalanced Classes*, *Numberanalytics.com*, https://www.numberanalytics.com/blog/f1-score-imbalanced-classes-guide, accessed 10 January 2026.

MLJourney (2025) *Decision Tree in Machine Learning: How They Work + Examples - ML Journey*, *ML Journey*, https://mljourney.com/decision-tree-in-machine-learning-how-they-work-examples/, accessed 10 January 2026.

scikit-learn (2019) *StandardScaler*, *scikit-learn.org*, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html, accessed 10 January 2026.

Song P (2025) *Cross Validation Strategies for Imbalanced Datasets - ML Journey*, *ML Journey*, https://mljourney.com/cross-validation-strategies-for-imbalanced-datasets/, accessed 9 January 2026.

Vinmec (2024) *Phân biệt u lành tính và u ác tính*, *Vinmec International Hospital*, https://www.vinmec.com/vie/bai-viet/phan-biet-buou-lanh-tinh-va-buou-ac-tinh-vi, accessed 29 December 2025.

Wilcox RR (2003) *Winsorization - an overview | ScienceDirect Topics*, *www.sciencedirect.com*, https://www.sciencedirect.com/topics/mathematics/winsorization, accessed 10 January 2026.

Wolberg W, Mangasarian O, Street N and Street W (1995) *UCI Machine Learning Repository*, *archive.ics.uci.edu*, https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic, accessed 8 January 2026.